

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成24年 5月 22日現在

機関番号：32689

研究種目：若手研究（B）

研究期間：2010～2011

課題番号：22700048

研究課題名（和文） クラウド環境での機械学習アルゴリズムの高速分散実行機構

研究課題名（英文） A Framework for Machine Learning Algorithms on the Cloud

研究代表者

秋岡 明香（AKIOKA SAYAKA）

早稲田大学・IT研究機構・准教授

研究者番号：90333533

研究成果の概要（和文）：

本研究提案では、Twitter のような非常に高速な大規模時系列データ群をリアルタイムに機械学習アルゴリズムに入力、実行可能な分散実行機構の構築を目指した。しかし実際には、クラウド等の十分に計算資源が確保可能な環境では、パイプライン化により数十台の計算機により入力データをほぼ処理することが可能であることが明確になった。GPGPU などを含めた、より並列度が高い環境を対象とした分散実行機構実現のためには、データ入力速度の高速化が重要であるとの知見を得た。

研究成果の概要（英文）：

This proposal originally tried to develop a framework, which enables to run machine learning algorithms in parallel on the computational cloud with the full input from the very fast and huge data source such as Twitter. However, several experiments in this project have revealed that a couple dozen CPUs in the cloud are enough to process all the inputs in a pipeline. For the higher level of the parallelism, such as the execution on GPGPUs, the drastic speed-up of the input is unavoidable.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
22年度	2,284,638	690,000	2,974,638
23年度	1,699,427	510,000	2,209,427
年度			
年度			
年度			
総計	3,984,065	1,200,000	5,184,065

研究分野：総合領域

科研費の分科・細目：情報学・計算機システム・ネットワーク

キーワード：機械学習、並列分散処理、クラウドコンピューティング

1. 研究開始当初の背景

Apache Mahoutプロジェクト[1,2]（以下、Mahout）は、機械学習アルゴリズムのクラウド環境での実装の代表であり、スケーラブルな機械学習アルゴリズムライブラリの提供を目指している。一部の機械学習アルゴリズムは、Apache Hadoopライブラリ[3]（以

下、Hadoop）の使用により分散実行が可能である。Hadoopは、クラウド環境におけるアプリケーションの代表的な並列分散実行モデルのひとつであるMapReduce[4]の実装である。

しかし、既存の機械学習アルゴリズムは必ずしもMapReduceモデルに合致せず、

Mahout で分散実行が可能となる機械学習アルゴリズムは一部にすぎない。Gillick らは、機械学習アルゴリズムを MapReduce モデルで分散実行した場合の問題点を次の通り指摘している[5]。

- (1) 機械学習アルゴリズムの大半は、学習状態や学習パラメタの全体での共有が必要であり、既存の MapReduce フレームワークは効果的なデータ共有方法を提供していない。
→学習状態や学習パラメタの共有（ブロードキャスト等）がボトルネックとなる。
- (2) Hadoop では、タスクと処理データをひとつにパッケージする必要がある。一方で、機械学習では、ひとつの入力ファイルに対して複数の異なる処理を行なうことや、先行タスクが出力した複数のファイルを組み合わせる処理することが多い。
→各タスクが任意の複数データに効率的にアクセスできる方法が必要である。

申請者は、これまで以下の研究を通して、本研究に応用可能な成果を得ている。

- (1) 分散環境での負荷分散手法やグリッドスケジューラの提案
→クラウド環境におけるスケジューラ作成に応用可能
- (2) グリッド環境における CPU やネットワーク負荷の予測手法の提案
→クラウド環境における資源状態の予測手法に応用可能
- (3) マルチコアシステムでのアプリケーションのデータアクセスパターンに基づくキャッシュアーキテクチャの提案
→機械学習アルゴリズムのデータアクセスパターンの抽出とデータ配置手法に応用可能
- (4) 機械学習によるアプリケーション異常検出手法の提案、および機械学習による商品レコメンデーション手法の提案
→機械学習アルゴリズムについての知識と経験

以上の成果を発展させることで、クラウド環境の資源状態に基づき、機械学習におけるデータアクセスパターンに適したデータとタスクの配置を行なうスケジューラが実現可能となり、機械学習アルゴリズムを高速に分散実行することが可能となる。

参考文献

- [1] Apache Mahout, <http://lucene.apache.org/mahout>.
- [2] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, “Map-Reduce for Machine Learning on Multicore”, Neural Information Processing

Systems Conference (NIPS19), 2007.

- [3] Apache Hadoop, <http://hadoop.apache.org/>.
- [4] J. Dean, and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Sixth Symposium on Operating System Design and Implementation (OSDI’04), 2004.
- [5] D. Gillick, A. Faria, and J. DeNero, “MapReduce: Distributed Computing for Machine Learning”, Department of Computer Science, UC Berkeley, 2006.

2. 研究の目的

本研究では、クラウドコンピューティング環境（以下、クラウド環境）において、機械学習アルゴリズムを高速に分散実行する機構を実現する。

機械学習アルゴリズムを分散実行する場合の主要なボトルネックは、学習状態や学習パラメタの共有、および入力データ群へのアクセスである。提案する分散実行機構は、機械学習アルゴリズムのデータアクセスパターン、およびクラウド環境の計算資源負荷情報に基づいたタスクおよびデータの最適な配置を行ない、高速な分散実行環境を提供する。また、クラウド環境の代表的な分散モデルである MapReduce モデルが適用不可能な機械学習アルゴリズムについても、分散実行可能なライブラリ実装を提供する。

3. 研究の方法

研究の柱は、1)機会学習アルゴリズムのアクセスパターン解析、2)負荷予測やアクセスパターン解析結果を用いるスケジューラの作成、3)データアクセスの高速化である。研究を効果的に進めるために、1年目は MapReduce モデルが適用可能な機械学習アルゴリズムを中心に、翌年は MapReduce モデルが適さない機械学習アルゴリズムを中心に、上記3テーマの研究を進める。また、負荷予測、アクセスパターン解析、スケジューラについては、過去の研究成果も応用する。

研究進捗管理、および他の研究者との意見交換のため、各年度で 1)機械学習アルゴリズムのアクセスパターン解析とボトルネック抽出、2)機械学習アルゴリズムの高速分散実行を可能とするスケジューラ、3)レプリカ等を用いたデータアクセスの高速化、さらに 2年目には 4)MapReduce モデルによらない機械学習の分散実行ライブラリについての学会発表を予定している。

本課題での実施予定項目

- (1) ネットワークを中心としたクラウド環境の負荷予測手法の開発・実装
- (2) クラウド環境におけるスケジューラの

プロトタイプ作成

- (3) 負荷予測結果に基づくスケジューリングアルゴリズムの開発と実装
- (4) 機械学習アルゴリズム (MapReduce 適用可能) のデータアクセスパターン解析
- (5) データアクセスパターンの解析結果に基づくスケジューリングアルゴリズムを用いる機械学習アルゴリズム (MapReduce 適用可能) の高速化
- (6) 機械学習アルゴリズム (MapReduce 適用不可) の分散実装
- (7) 機械学習アルゴリズム (MapReduce 適用不可) のデータアクセスパターン解析
- (8) レプリカ等の利用によるデータアクセスの高速化手法を用いたスケジューリングアルゴリズムの開発・実装と性能評価

平成 22 年度の予定

平成 22 年度は、実施予定項目のうち、1~5 を実施する。つまり、クラウド環境の計算資源負荷予測、および MapReduce モデルで分散実行が可能な機械学習アルゴリズムのデータアクセスパターンを考慮したスケジューラと性能評価を行なう。なお、MapReduce モデルで分散実行が可能な機械学習アルゴリズムとしては、K-means 法、遺伝的アルゴリズム、Mean-shift 法などを高速化の対象とする予定である。

平成 23 年度の予定

平成 23 年度は、実施予定項目のうち、6~8 を実施する。つまり、以下の成果物を予定している。

- ・ MapReduce モデルでの分散実行が適切でない機械学習アルゴリズムを分散化するライブラリ
- ・ 上記機械学習アルゴリズムのデータアクセスパターンに対応したスケジューラと性能評価
- ・ レプリカ利用等によるデータアクセスの高速化手法

なお、MapReduce モデルでの分散実行が適切でない機械学習アルゴリズムとしては、ニューラルネットワーク、隠れマルコフモデル、ブースティングなどを高速化の対象とする予定である。

4. 研究成果

個別の機械学習アルゴリズムの並列化を行なう前段階として、最初に入力データの速度に対して、並列化による処理の高速化をどの程度実現しなければ入力データを失うことになるかを調べるために、次のような実験を行なった。

2010 年時点での Twitter のデータ量は、1 日に 100,000,000 ツイート (1 ツイートは 140 文字) であると言われていたことから、実際

のツイートをサンプリングにより収集し、収集したツイート群をシードとして、毎秒 1000 ツイートを生成するデータストリームを生成し、これを入力として実験を行なった。

実験では、入力された各ツイートに対して形態素解析を行ない、出現する名詞やユーザー名の出現頻度を調べる処理を、クラウド環境でパイプライン化し、使用した計算ノード数と処理しきれずに破棄した入力データ (ツイート) 数を調べた。なおここで、入力データを処理しきれずに破棄する場合は、前のデータに対する処理に時間を要しており、パイプライン処理が滞っているが、計算資源の不足により、これ以上の計算資源を投入してパイプライン段数を増やすことができない状況を指す。

この実験で用いたクラウド環境は、IBM 社の IBM Computing on Demand (CoD) で、計算ノードは合計で 52 ノード、パイプライン段数を決定したスクを割り当てるリソースマネージャとして別途 1 ノード使用した。各計算ノードとリソースマネージャのスペックを Table 1 に示す。なお、ツイートの解析モジュールやリソースマネージャ等はすべて Java による実装を行った。

Table 1 各計算ノードおよびリソースマネージャのスペック一覧

Node Type	Num. of nodes	CPU+Memory
Node Type A	32 nodes	Xeon X5550+32GB
Node Type B	14 nodes	Xeon X5450+8GB
Node Type C	6 nodes	Xeon X5430+8GB
Resource Mngr.	1 node	Xeon E5520+24GB

Figure 1 に、上記の解析処理をシリアル実行した場合の入力データ廃棄率と、同等の解析処理を提案手法によりパイプライン処理した場合の入力データ廃棄率を比較した結果を示す。ここで、入力データの廃棄率 $R_{processed}$ は次のように定義する。

$$R_{processed} = 100(1 - \frac{n_{processed}}{n_{total}})$$

$n_{processed}$ は解析対象となったツイートの総数、 n_{total} は入力されたツイートの総数を表す。

さらに Figure 1 において、x 軸はひとつの計算ノードに一度に割り当てられるツイート数である。理想的には、1 ツイートを 1 計算ノードに割り当てることが望ましいが、今回の実験では計算資源の総量が 52 ノードと決まっており、全 52 ノードを用いても入力データを破棄することなく理想的な資源割り当てを行なうことができなかったため、1 計算ノードに割り当てるツイート数を様々に変更して実験を行なった。

Figure 1 に示す通り、シリアル実行では 1 計算資源に割り当てるツイート数を 60 ツイートにまで増やしても 80% 以上の入力データ

を破棄する結果になる一方、提案するパイプライン実行では8ツイート以上をまとめて1つの計算資源に割り当てることにより、入力データを100%処理することができることがわかった。

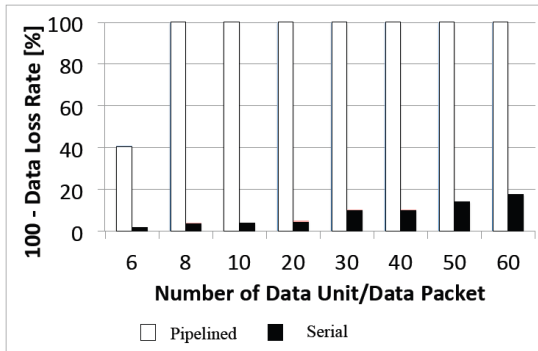


Figure 1 シリアル実行時（黒）とクラウド環境で最大52計算ノードを用いてパイプライン実行時（白）に、Twitterの全入力と同等スピードのデータを処理した場合の処理率の比較。

こうした実験の結果により、fork-join型の並列分散処理に依存せずにパイプラインで段数を増やし、入力データのバッファに相当する効果を得ることが出来れば、現実的な高速データストリームをすべて解析処理する基盤が実現可能であることがわかった。

こうした知見をさらに発展させて、GPGPUなどのより並列度の高い環境での機械学習

の並列実行実現についても検討を行ったが、こうした環境では入力データ速度がボトルネックとなり、解析処理自体が十分な並列度を有していないことがわかった。今後は、クラウドのみでなく、より広範な並列分散処理環境に対応すべく、入力データ速度のボトルネック解消手法について、研究を進める予定である。

5. 主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔学会発表〕（計1件）

1. Sayaka Akioka, Hayato Yamana, Yoichi Muraoka, “Dynamic Stream Allocation with the Discrepancy between Data Access Time and CPU Usage Time”, The 4th International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA2012), 2012.

〔その他〕

1. 秋岡明香, “インターネットの公開情報を総合的に用いた社会分析基盤の構築”, 早稲田大学理工展サイエンスカフェにおける講演, 2011年11月.

6. 研究組織

(1) 研究代表者

秋岡明香（早稲田大学 IT 研究機構主任研究員）

研究者番号：90333533